

VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	

LL	IIIIII	BBBBBBBB	FFFFFFFF	IIIIII	LL	PPPPPPPP	RRRRRRRR	000000			
LL	IIIIII	BBBBBBBB	FFFFFFFF	IIIIII	LL	PPPPPPPP	RRRRRRRR	000000			
LL	II	BB	BB	FF	II	PP	PP	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	PP	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	PP	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	PP	RR	RR	00	00
LL	II	BBBBBBBB	FFFFFFFF	II	II	PPPPPPPP	RRRRRRRR	RR	RR	00	00
LL	II	BBBBBBBB	FFFFFFFF	II	II	PPPPPPPP	RRRRRRRR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LL	II	BB	BB	FF	II	PP	RR	RR	RR	00	00
LLLLLLLLLL	IIIIII	BBBBBBBB	FF	IIIIII	LLLLLLLLLL	PP	RR	RR	RR	000000	00
LLLLLLLLLL	IIIIII	BBBBBBBB	FF	IIIIII	LLLLLLLLLL	PP	RR	RR	RR	000000	00

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LI          II
LL          II
LLLLLLLLLLL IIIIII
LLLLLLLLLLL IIIIII

SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSSS
SSSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

LIE
VO4

...

.....

.....

⋮

SECRET
.....


```
0001 0 %TITLE 'LIB$SET_FILE_PROT - Set file protection'
0002 0 MODULE LIB$SET_FILE_PROT ( ! Set file protection
0003 0 IDENT = 'V04-000' ! File: LIBFILPRO.B32 Edit: 1-001
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: General Utility Library
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This routine sets the file protection of a file.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Martin L. Jack, CREATION DATE: 23-Dec-1981
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original from LIBACP.B32. MLJ 23-Dec-1981
0045 1 --
0046 1
```

```

: 48      0047 1 %SBTTL 'Declarations'
: 49      0048 1
: 50      0049 1 | SWITCHES:
: 51      0050 1 |
: 52      0051 1 |
: 53      0052 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
: 54      0053 1 |
: 55      0054 1 |
: 56      0055 1 | LINKAGES:
: 57      0056 1 |
: 58      0057 1 LINKAGE
: 59      0058 1 | LINKAGE_JSB_2_2 = JSB(REGISTER=0;REGISTER=1,REGISTER=2);
: 60      0059 1 |
: 61      0060 1 | TABLE OF CONTENTS:
: 62      0061 1 |
: 63      0062 1 |
: 64      0063 1 FORWARD ROUTINE
: 65      0064 1 | LIB$SET_FILE_PROT; | Set file protection
: 66      0065 1 |
: 67      0066 1 |
: 68      0067 1 | INCLUDE FILES:
: 69      0068 1 |
: 70      0069 1 |
: 71      0070 1 LIBRARY 'SYS$LIBRARY:LIB'; | System symbols
: 72      0071 1 |
: 73      0072 1 |*! REQUIRE 'RTLIN:RTLPSECT'; | Define PSECT declarations macros
: 74      0073 1 |
: 75      0074 1 |
: 76      0075 1 | MACROS:
: 77      0076 1 |
: 78      0077 1 | NONE
: 79      0078 1 |
: 80      0079 1 | EQUATED SYMBOLS:
: 81      0080 1 |
: 82      0081 1 | NONE
: 83      0082 1 |
: 84      0083 1 | FIELDS:
: 85      0084 1 |
: 86      0085 1 | NONE
: 87      0086 1 |
: 88      0087 1 | PSECTS:
: 89      0088 1 |
: 90      0089 1 |*! DECLARE_PSECTS (LIB); | Declare PSECTs for LIB$ facility
: 91      0090 1 | PSECT
: 92      0091 1 | CODE = _LIB$CODE (READ, NOWRITE, EXECUTE, SHARE, PIC, ADDRESSING_MODE (WORD_RELATIVE)),
: 93      0092 1 | PLIT = _LIB$CODE (READ, NOWRITE, EXECUTE, SHARE, PIC, ADDRESSING_MODE (WORD_RELATIVE)),
: 94      0093 1 | OWN = _LIB$DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC, ADDRESSING_MODE (LONG_RELATIVE)),
: 95      0094 1 | GLOBAL = _LIB$DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC, ADDRESSING_MODE (LONG_RELATIVE));
: 96      0095 1 |
: 97      0096 1 | OWN STORAGE:
: 98      0097 1 |
: 99      0098 1 | NONE
: 100     0099 1 |
: 101     0100 1 | EXTERNAL REFERENCES:
: 102     0101 1 |
: 103     0102 1 |
: 104     0103 1 EXTERNAL ROUTINE
```


LIB\$SET_FILE_PROT - Set file protection
V04-000 Declarations

L 5
16-Sep-1984 02:26:57 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:34:28 [VMSLIB.SRC]LIBFILPRO.B32;1

Page 3
(2)

:	105	0104	1	LIB\$ANALYZE_SDESC_R2:	LINKAGE-JSB_2_2,	! Analyze descriptor
:	106	0105	1	LIB\$FREE_EF,	!	Deallocate an event flag
:	107	0106	1	LIB\$GET_EF,	!	Allocate an event flag
:	108	0107	1	SYSS\$SETDFPROT;	!	Read/set default file protection
:	109	0108	1			
:	110	0109	1	EXTERNAL LITERAL	!	Completion status codes
:	111	0110	1	LIB\$INVARG,	!	Invalid argument
:	112	0111	1	LIB\$INVFILSPE;	!	Invalid file specification

```
114 0112 1 %SBTTL 'LIB$SET_FILE_PROT - Set file protection'
115 0113 1 GLOBAL ROUTINE LIB$SET_FILE_PROT (
116 0114 1     FILE_SPEC,
117 0115 1     ENABLE_MASK,
118 0116 1     VALUE_MASK,
119 0117 1     ACTUAL_MASK
120 0118 1 ) =
121 0119 1
122 0120 1 ++
123 0121 1 FUNCTIONAL DESCRIPTION:
124 0122 1
125 0123 1     This routine sets the file protection of a file.
126 0124 1
127 0125 1 CALLING SEQUENCE:
128 0126 1
129 0127 1     ret_status.wlc.v = LIB$SET_FILE_PROT (file-spec.rt.dx,
130 0128 1     [enable-mask.rwu.r], [value-mask.rwu.r], [actual-mask.wwu.r])
131 0129 1
132 0130 1 FORMAL PARAMETERS:
133 0131 1
134 0132 1     FILE_SPEC      Address of a descriptor for the file specification.
135 0133 1     This string is a standard RMS file specification; it
136 0134 1     must not contain a node name nor wild card characters;
137 0135 1     it must reference a disk device. The string must be
138 0136 1     no longer than 255 characters.
139 0137 1
140 0138 1     ENABLE_MASK    Address of a word containing a mask to specify the bits
141 0139 1     of the file protection to be modified. Bits of the
142 0140 1     file protection corresponding to set bits in
143 0141 1     enable-mask are set to the value of the corresponding
144 0142 1     bit of the value-mask parameter; bits of the file
145 0143 1     protection corresponding to clear bits in enable-mask
146 0144 1     are not modified. This is an optional parameter. The
147 0145 1     default is a mask of all one bits, which results in
148 0146 1     changing all bits of the file protection.
149 0147 1
150 0148 1     VALUE_MASK     Address of a word containing a mask to specify the
151 0149 1     value of modified bits of the file protection. Bits of
152 0150 1     the file protection corresponding to set bits in
153 0151 1     enable-mask are set to the value of the corresponding
154 0152 1     bit of the value-mask parameter. This is an optional
155 0153 1     parameter. The default is the current process default
156 0154 1     file protection.
157 0155 1
158 0156 1     ACTUAL_MASK    Address of a word that receives the file protection
159 0157 1     value actually set. This is an optional output
160 0158 1     parameter. It is guaranteed to be stored only if the
161 0159 1     routine value is true.
162 0160 1
163 0161 1     The format of ENABLE_MASK, VALUE_MASK, and ACTUAL_MASK is:
164 0162 1
165 0163 1         1 1 1 1 1 1
166 0164 1         5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
167 0165 1         +-----+-----+-----+-----+
168 0166 1         |World|Group|Owner|System|
169 0167 1         |D E W R|D E W R|D E W R|D E W R|
170 0168 1         +-----+-----+-----+-----+
```


LIB\$SET_FILE_PROT - Set file protection
V04-000 LIB\$SET_FILE_PROT - Set file protection

N 5
16-Sep-1984 02:26:57
14-Sep-1984 13:34:28

VAX-11 Bliss-32 V4.0-742
[VMSLIB.SRC]LIBFILPRO.B32;1

Page 5
(3)

```

171 0169 1
172 0170 1      Set bits deny access and clear bits grant access.
173 0171 1
174 0172 1      IMPLICIT INPUTS:
175 0173 1
176 0174 1      NONE
177 0175 1
178 0176 1      IMPLICIT OUTPUTS:
179 0177 1
180 0178 1      NONE
181 0179 1
182 0180 1      COMPLETION STATUS:
183 0181 1
184 0182 1      SSS_NORMAL      Normal successful completion
185 0183 1
186 0184 1      LIB$_INVARG      Required argument omitted, or file-spec longer than
187 0185 1      255 characters
188 0186 1
189 0187 1      LIB$_INVFILSPE  File specification contained a node name or wildcard or
190 0188 1      did not reference a disk device
191 0189 1
192 0190 1      LIB$ANALYZE_SDESC errors
193 0191 1      $PARSE errors
194 0192 1      $SEARCH errors
195 0193 1      $ASSIGN errors
196 0194 1      LIB$GET_EF errors
197 0195 1      $QIO errors
198 0196 1      $DASSGN errors
199 0197 1      LIB$FREE_EF errors
200 0198 1
201 0199 1      SIDE EFFECTS:
202 0200 1
203 0201 1      File protection of file modified.
204 0202 1
205 0203 1      --
206 0204 1
207 0205 2      BEGIN
208 0206 2      LOCAL
209 0207 2      FAB:          $FAB_DECL,      ! FAB for $PARSE
210 0208 2      NAM:          $NAM_DECL,      ! NAM block for $PARSE
211 0209 2      ESA_BUFFER:  VECTOR[NAM$C_MAXRSS,BYTE], ! Expanded string area
212 0210 2      TEMP_DESC:   BLOCK[DSC$K_5_BLN,BYTE], ! Utility descriptor
213 0211 2      FPRO:        BLOCK[ATR$S_FPRO,BYTE]
214 0212 2      VOLATILE,    ! File protection
215 0213 2      FIB:          BLOCK[FIB$C_LENGTH,BYTE], ! FIB
216 0214 2      FIB_DESC:    VECTOR[2],      ! Descriptor for FIB
217 0215 2      ATR:          BLOCKVECTOR[2,8,BYTE], ! Attribute descriptors
218 0216 2      IOSB:         VECTOR[4,WORD],  ! I/O status block
219 0217 2      CHANNEL:     WORD,            ! Channel number
220 0218 2      EFN,          ! Event flag number
221 0219 2      LOCAL_ENABLE: WORD,            ! Actual or defaulted ENABLE_MASK
222 0220 2      LOCAL_VALUE: WORD,            ! Actual or defaulted VALUE_MASK
223 0221 2      STATUS_1,     ! Status return
224 0222 2      STATUS_2,     ! Status return
225 0223 2      STATUS_3,     ! Status return
226 0224 2      STATUS_4,     ! Status return
227 0225 2      STATUS_5,     ! Status return
```



```
228      0226 2      STATUS_6,      ! Status return
229      0227 2      STATUS_7,      ! Status return
230      0228 2      FINAL_STATUS;    ! Status return
231      0229 2 LABEL
232      0230 2      PROCESS;          ! Block exited when processing complete
233      0231 2 BUILTIN
234      0232 2      ACTUALCOUNT,      ! Return number of arguments
235      0233 2      NULLPARAMETER;      ! Test if parameter specified
236      0234 2
237      0235 2 !+
238      0236 2 !- Ensure that the required parameter is present.
239      0237 2 !-
240      0238 2
241      0239 2 IF ACTUALCOUNT() EQL 0 THEN RETURN LIB$_INVARG;
242      0240 2
243      0241 2 !+
244      0242 2 !- Initialize RMS structures required to do a $PARSE and $SEARCH.
245      0243 2 !-
246      0244 2
247      P 0245 2 $FAB_INIT(FAB=FAB,
248      0246 2     NAM=NAM);
249      P 0247 2 $NAM_INIT(NAM=NAM,
250      0248 2     ESA=ESA_BUFFER,
251      0249 2     ESS=NAM$_C_MAXRSS);
252      0250 2
253      0251 2 !+
254      0252 2 !- Analyze the input descriptor and set up the FAB filename descriptor.
255      0253 2 !-
256      0254 2
257      0255 2 BEGIN ! block to use output registers
258      0256 2 REGISTER
259      0257 2     R1 = 1;
260      0258 2     R2 = 2;
261      0259 2
262      0260 2 STATUS_1 = LIB$ANALYZE_SDESC_R2(.FILE_SPEC; R1, R2);
263      0261 2 IF NOT .STATUS_1 THEN RETURN .STATUS_1;
264      0262 2 IF .R1 GTRU 255 THEN RETURN LIB$_INVARG;
265      0263 2 FAB[FAB$_FNS] = .R1;
266      0264 2 FAB[FAB$_FNA] = .R2;
267      0265 2 END; ! block to use output registers
268      0266 2
269      0267 2 !+
270      0268 2 !- Parse the file specification to obtain the expanded name string.
271      0269 2 !-
272      0270 2
273      0271 2 STATUS_2 = $PARSE(FAB=FAB);
274      0272 2 IF NOT .STATUS_2 THEN RETURN .STATUS_2;
275      0273 2
276      0274 2 !+
277      0275 2 !- Perform various error checks on the file specification. It must not have a
278      0276 2 !- node name, must not contain wildcards, and must reference a disk device.
279      0277 2 !-
280      0278 2
281      0279 2 IF
282      0280 2     (.NAM[NAM$_FNB] AND (NAM$_WILDCARD OR NAM$_NODE)) NEQ 0 OR
283      0281 2     NOT .BLOCK[FAB[FAB$_DEV], DEV$_RND; ,BYTE]
284      0282 2 THEN
```



```

285      0283 3 BEGIN
286      0284 3
287      0285 3      !+
288      0286 3      ! If the string contained a wildcard or a node name, internal RMS resources
289      0287 3      ! have been consumed by $PARSE. Execute another $PARSE using the same FAB
290      0288 3      ! on a null string to release these resources.
291      0289 3      !-
292      0290 3
293      0291 3      FAB[FAB$B_FNS] = 0;
294      0292 3      $PARSE(FAB=FAB);
295      0293 3      RETURN LIB$_INVFILSPE;
296      0294 3      END;
297      0295 3
298      0296 3      !+
299      0297 3      ! Perform a $SEARCH to get the file identification of the file.
300      0298 3      !-
301      0299 3
302      0300 3      STATUS_3 = $SEARCH(FAB=FAB);
303      0301 3      IF NOT .STATUS_3 THEN RETURN .STATUS_3;
304      0302 3
305      0303 3      !+
306      0304 3      ! Set up the FIB.
307      0305 3      !-
308      0306 3
309      0307 3      CH$FILL(0, FIB$C_LENGTH, FIB);
310      0308 3      FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NOREAD OR FIB$M_NOWRITE;
311      0309 3      FIB[FIB$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
312      0310 3      FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
313      0311 3      FIB[FIB$W_FID_RVN] = .NAM[NAM$W_FID_RVN];
314      0312 3
315      0313 3      !+
316      0314 3      ! Set up the FIB descriptor.
317      0315 3      !-
318      0316 3
319      0317 3      FIB_DESC[0] = FIB$C_LENGTH;
320      0318 3      FIB_DESC[1] = FIB;
321      0319 3
322      0320 3      !+
323      0321 3      ! Set up the attribute list.
324      0322 3      !-
325      0323 3
326      0324 3      ATR[0, ATR$W_TYPE] = ATR$C_FPRO;          ! File protection
327      0325 3      ATR[0, ATR$W_SIZE] = ATR$S_FPRO;
328      0326 3      ATR[0, ATR$L_ADDR] = FPRO;
329      0327 3      ATR[1, 0,0,32,0] = 0;                    ! End of list
330      0328 3
331      0329 3      !+
332      0330 3      ! Assign a channel to the device.
333      0331 3      !-
334      0332 3
335      0333 3      TEMP_DESC[DSC$W_LENGTH] = .NAM[NAM$B_DEV];
336      0334 3      TEMP_DESC[DSC$A_POINTER] = .NAM[NAM$C_DEV];
337      0335 3      STATUS_4 = $ASSIGN(DEVNAM=TEMP_DESC, CHAN=CHANNEL);
338      0336 3      IF NOT .STATUS_4 THEN RETURN .STATUS_4;
339      0337 3
340      0338 3      !+
341      0339 3      ! Allocate an event flag.
```



```
342 0340 2 !-
343 0341 2
344 0342 2 STATUS_5 = LIB$GET_EF(EFN);
345 0343 2 IF NOT .STATUS_5
346 0344 2 THEN
347 0345 2 BEGIN
348 0346 2     $DASSGN(CHAN=.CHANNEL);
349 0347 2     RETURN .STATUS_5;
350 0348 2 END;
351 0349 2
352 0350 2 !+
353 0351 2 ! Beginning of block that is exited when processing is complete. FINAL_STATUS
354 0352 2 ! contains the status to be returned to caller.
355 0353 2 !-
356 0354 2
357 0355 2 PROCESS: BEGIN
358 0356 2
359 0357 2 !+
360 0358 2 ! Evaluate the enable-mask.
361 0359 2 !-
362 0360 2
363 0361 2 LOCAL_ENABLE = -1; ! Assume default
364 0362 2 IF NOT NULLPARAMETER(2)
365 0363 2 THEN
366 0364 2     LOCAL_ENABLE = .(.ENABLE_MASK)<0,16>;
367 0365 2
368 0366 2 !+
369 0367 2 ! If the enable-mask does not specify a change to all bits of the existing
370 0368 2 ! file protection, read attributes of the file.
371 0369 2 !-
372 0370 2
373 0371 2 IF .LOCAL_ENABLE NEQ 65535
374 0372 2 THEN
375 0373 2 BEGIN
376 0374 2     FINAL_STATUS = $QIOW(
377 0375 2     FUNC=IOS$ ACCESS,
378 0376 2     CHAN=.CHANNEL,
379 0377 2     EFN=.EFN,
380 0378 2     IOSB=IOSB,
381 0379 2     P1=FIB_DESC,
382 0380 2     P5=ATTR);
383 0381 2     IF .FINAL_STATUS THEN FINAL_STATUS = .IOSB[0];
384 0382 2     IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
385 0383 2 END;
386 0384 2
387 0385 2 !+
388 0386 2 ! Evaluate the value-mask.
389 0387 2 !-
390 0388 2
391 0389 2 IF NULLPARAMETER(3)
392 0390 2 THEN
393 0391 2     SYS$SETDFPROT(0, LOCAL_VALUE) ! Read process default
394 0392 2 ELSE
395 0393 2     LOCAL_VALUE = .(.VALUE_MASK)<0,16>;
396 0394 2
397 0395 2 !+
398 0396 2 ! Evaluate the new protection value.
```



```
399 0397 3 !-
400 0398
401 0399 FPRO = (.FPRO AND NOT .LOCAL_ENABLE) OR (.LOCAL_VALUE AND .LOCAL_ENABLE);
402 0400
403 0401 !+
404 0402 ! If requested, return the new protection value.
405 0403 !-
406 0404
407 0405 IF NOT NULLPARAMETER(4)
408 0406 THEN
409 0407     (.ACTUAL_MASK)<0,16> = .FPRO;
410 0408
411 0409 !+
412 0410 ! Rewrite the file protection.
413 0411 !-
414 0412
415 P 0413 FINAL STATUS = $QIOW(
416 P 0414     FUNC=IOS$ MODIFY,
417 P 0415     CHAN=.CHANNEL,
418 P 0416     EFN=.EFN,
419 P 0417     IOSB=IOSB,
420 P 0418     P1=FIB_DESC,
421 0419     P5=ATR);
422 0420 IF .FINAL_STATUS THEN FINAL STATUS = .IOSB[0];
423 0421 IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
424 0422 FINAL_STATUS = SS$NORMAL;
425 0423
426 0424 !+
427 0425 ! End of block that is exited when processing is complete. FINAL_STATUS
428 0426 ! contains the status that is to be returned to caller.
429 0427 !-
430 0428
431 0429 END; ! of block PROCESS
432 0430
433 0431 !+
434 0432 ! Deassign the channel and deallocate the event flag.
435 0433 !-
436 0434
437 0435 STATUS_6 = $DASSGN(CHAN=.CHANNEL);
438 0436 STATUS_7 = LIB$FREE_EF(EFN);
439 0437 IF NOT .STATUS_7 THEN RETURN .STATUS_7;
440 0438 IF NOT .STATUS_6 THEN RETURN .STATUS_6;
441 0439
442 0440 !+
443 0441 ! Return the status.
444 0442 !-
445 0443
446 0444 RETURN .FINAL_STATUS;
447 0445 1 END;
```

! End of routine LIB\$SET_FILE_PROT

```
.TITLE LIB$SET_FILE_PROT LIB$SET_FILE_PROT - Set file
      protection
.IDENT \V04-000\
.EXTRN LIB$ANALYZE_SDESC R2
.EXTRN LIB$FREE_EF, LIB$GET_EF
```



```
                                OFFC 00000
0050 8F 00 58 00000000G 00 9E 00002
      57 00000000G 00 9E 00009
      56 00000000G 00 9E 00010
      5E FDD8 CE 9E 00017
      6C 95 0001C
      4F 13 0001E
      6E 00 2C 00020
      B0 AD 00027
      5003 8F B0 00029
      C6 AD 02 90 0002F
      CF AD 02 90 00033
      D8 AD FF50 CD 9E 00037
0060 8F 00 6E 00 2C 0003D
      FF50 CD 00044
      6002 8F B0 00047
      FF5A CD 01 8E 0004E
      FF5C CD 78 AE 9E 00053
      50 04 AC D0 00059
      00000000G 00 16 0005D
      4A 50 E9 00063
      000000FF 8F 51 D1 00066
      50 00000000G 08 1B 0006D
      8F 8F D0 0006F 1$:
      E4 AD 51 90 00077 2$:
      DC AD 52 D0 0007B
      B0 AD 9F 0007F
      66 01 FB 00082
      79 50 E9 00085
      0201 8F 85 AD B3 00088
      11 F3 AD 05 12 0008E
      E4 AD 04 E0 00090
      B0 AD 94 00095 3$:
      66 01 FB 00098
      50 00000000G 8F D0 0009E
      B0 AD 04 000A5
      00000000G 00 AD 9F 000A6 4$:
      4E 01 FB 000A9
      6E 00 50 E9 000B0 5$:
      2C AE 0501 8F 3C 000BC
      30 AE FF74 CD D0 000C2
      34 AE FF78 CD B0 000C8
      24 AE 40 8F 9A 000CE
      28 AE 2C AE 9E 000D3
      14 AE 00160002 8F D0 000D8
```

```
.EXTRN SYSS$SETDFPROT, LIB$ _INVARG
.EXTRN LIB$ _INVFILSPE, SYSS$PARSE
.EXTRN SYSS$SEARCH, SYSS$ASSIGN
.EXTRN SYSS$DASSGN, SYSS$QIOW

.PSECT _LIB$CODE,NOWRT, SHR, PIC,2

.ENTRY LIB$SET_FILE_PROT, Save R2,R3,R4,R5,R6,R7,- 0113
      R8,R9,R10,R11
      MOVAB SYSS$QIOW, R8
      MOVAB SYSS$DASSGN, R7
      MOVAB SYSS$PARSE, R6
      MOVAB -552(SP), SP
      TSTB (AP) 0239
      BEQL 1$
      MOVCS #0, (SP), #0, #80, $RMS_PTR 0246

      MOVW #20483, $RMS_PTR
      MOVW #2, $RMS_PTR+22
      MOVW #2, $RMS_PTR+31
      MOVAB NAM, $RMS_PTR+40
      MOVCS #0, (SP), #0, #96, $RMS_PTR 0249

      MOVW #24578, $RMS_PTR
      MNEGB #1, $RMS_PTR+10
      MOVAB ESA_BUFFER, $RMS_PTR+12
      MOVL FILE_SPEC, R0 0260
      JSB LIB$ANALYZE_SDESC_R2
      BLBC STATUS_1, 5$ 0261
      CMPL R1, #255 0262
      BLEQU 2$
      MOVL #LIB$_INVARG, R0
      RET
      MOVW R1, FAB+52 0263
      MOVL R2, FAB+44 0264
      PUSHAB FAB 0271
      CALLS #1, SYSS$PARSE
      BLBC STATUS_2, 6$ 0272
      BITW NAM+52, #513 0280
      BNEQ 3$
      BBS #4, FAB+67, 4$ 0281
      CLRB FAB+52 0291
      PUSHAB FAB 0292
      CALLS #1, SYSS$PARSE
      MOVL #LIB$_INVFILSPE, R0 0293
      RET
      PUSHAB FAB 0300
      CALLS #1, SYSS$SEARCH
      BLBC STATUS_3, 6$ 0301
      MOVCS #0, (SP), #0, #64, FIB 0307

      MOVZWL #1281, FIB 0308
      MOVL NAM+36, FIB+4 0309
      MOVW NAM+40, FIB+8 0311
      MOVZBL #64, FIB_DESC 0317
      MOVAB FIB, FIB_DESC+4 0318
      MOVL #1441794, ATR 0325
```


LIB\$SET_FILE_PR LIB\$SET_FILE_PROT - Set file protection
V04-000 LIB\$SET_FILE_PROT - Set file protection

G 6
16-Sep-1984 02:26:57
14-Sep-1984 13:34:28

VAX-11 Bliss-32 V4.0-742
[VMSLIB.SRC]LIBFILPRO.B32;1

Page 11
(3)

18	AE	6C	AE	9E	000E0	MOVAB	FPRO, ATR+4	0326
		1C	AE	D4	000E5	CLRL	ATR+8	0327
70	AE	89	AD	9B	000E8	MOVZBW	NAM+57, TEMP_DESC	0333
74	AE	94	AD	D0	000ED	MOVL	NAM+68, TEMP_DESC+4	0334
			7E	7C	000F2	CLRQ	-(SP)	0335
		08	AE	9F	000F4	PUSHAB	CHANNEL	
		7C	AE	9F	000F7	PUSHAB	TEMP_DESC	
00000000G	00		04	FB	000FA	CALLS	#4, SYSS\$ASSIGN	
	01		50	E8	00101	BLBS	STATUS_4, 7\$	0336
				04	00104	RET		
00000000G	00	08	AE	9F	00105	PUSHAB	EFN	0342
	52		01	FB	00108	CALLS	#1, LIB\$GET_EF	
	09		50	D0	0010F	MOVL	R0, STATUS_5	
	7E		52	E8	00112	BLBS	STATUS_5, 3\$	0343
	67		6E	3C	00115	MOVZWL	CHANNEL, -(SP)	0346
			01	FB	00118	CALLS	#1, SYSS\$DAS\$GN	
	52		00D0	31	0011B	BRW	16\$	0347
	02		01	AE	0011E	MNEGW	#1, LOCAL_ENABLE	0361
			6C	91	00121	CMPB	(AP), #2	0362
			09	1F	00124	BLSSU	9\$	
		08	AC	D5	00126	TSTL	8(AP)	
			04	13	00129	BEQL	9\$	
FFFF	52	08	BC	B0	0012B	MOVW	@ENABLE_MASK, LOCAL_ENABLE	0364
	8F		52	B1	0012F	CMPW	LOCAL_ENABLE, #65535	0371
			2A	13	00134	BEQL	10\$	
			7E	D4	00136	CLRL	-(SP)	0380
		18	AE	9F	00138	PUSHAB	ATR	
			7E	7C	0013B	CLRQ	-(SP)	
			7E	D4	0013D	CLRL	-(SP)	
		38	AE	9F	0013F	PUSHAB	FIB_DESC	
			7E	7C	00142	CLRQ	-(SP)	
		2C	AE	9F	00144	PUSHAB	IOSB	
			32	DD	00147	PUSHL	#50	
	7E	28	AE	3C	00149	MOVZWL	CHANNEL, -(SP)	
		34	AE	DD	0014D	PUSHL	EFN	
	68		0C	FB	00150	CALLS	#12, SYSS\$QIOW	
	53		50	D0	00153	MOVL	R0, FINAL_STATUS	
	7C		53	E9	00156	BLBC	FINAL_STATUS, 15\$	0381
	53	0C	AE	3C	00159	MOVZWL	IOSB, FINAL_STATUS	
	75		53	E9	0015D	BLBC	FINAL_STATUS, 15\$	0382
	03		6C	91	00160	CMPB	(AP), #3	0389
			05	1F	00163	BLSSU	11\$	
		0C	AC	D5	00165	TSTL	12(AP)	
			0E	12	00168	BNEQ	12\$	
		04	AE	9F	0016A	PUSHAB	LOCAL_VALUE	0391
			7E	D4	0016D	CLRL	-(SP)	
00000000G	00		02	FB	0016F	CALLS	#2, SYSS\$SETDFPROT	
			05	11	00176	BRB	13\$	
	04	AE	0C	BC	B0	MOVW	@VALUE_MASK, LOCAL_VALUE	0393
	51		6C	AE	3C	MOVZWL	FPRO, R1	0399
	50		52	3C	0017D	MOVZWL	LOCAL_ENABLE, R0	
	51		50	CA	00184	BICL2	R0, RT	
	50		04	AE	3C	MOVZWL	LOCAL_VALUE, R0	
	54		52	3C	0018B	MOVZWL	LOCAL_ENABLE, R4	
	54		54	D2	0018E	MCOML	R4, R4	
	50		54	CA	00191	BICL2	R4, R0	
6C	AE		51	A9	00194	BISW3	R1, R0, FPRO	

LIB\$SET_FILE_PRO LIB\$SET_FILE_PROT - Set file protection
V04-000 LIB\$SET_FILE_PROT - Set file protection

H 6
16-Sep-1984 02:26:57
14-Sep-1984 13:34:28

VAX-11 Bliss-32 V4.0-742
[VMSLIB.SRC]LIBFILPRO.B32;1

Page 12
(3)

04	6C	91	00199	CMPB	(AP), #4	: 0405
	0A	1F	0019C	BLSSU	14\$	:
10	AC	D5	0019E	TSTL	16(AP)	:
	05	13	001A1	BEQL	14\$	:
10	BC	6C	AE B0 001A3	MOVW	FPRO, @ACTUAL_MASK	: 0407
	7E	D4	001AB	CLRL	-(SP)	: 0419
	18	AE	9F 001AA	PUSHAB	ATR	:
	7E	7C	001AD	CLRL	-(SP)	:
	7E	D4	001AF	CLRL	-(SP)	:
38	AE	9F	001B1	PUSHAB	FIB_DESC	:
	7E	7C	001B4	CLRL	-(SP)	:
2C	AE	9F	001B6	PUSHAB	IOSB	:
	36	DD	001B9	PUSHL	#54	:
7E	AE	3C	001BB	MOVZWL	CHANNEL, -(SP)	:
	34	AE	DD 001BF	PUSHL	EFN	:
68	0C	FB	001C2	CALLS	#12, SYSSQIOW	:
53	50	D0	001C5	MOVL	RO, FINAL_STATUS	:
0A	53	E9	001C8	BLBC	FINAL_STATUS, 15\$	: 0420
53	0C	AE	3C 001CB	MOVZWL	IOSB, FINAL_STATUS	:
03	53	E9	001CF	BLBC	FINAL_STATUS, 15\$	: 0421
53	01	D0	001D2	MOVL	#1, FINAL_STATUS	: 0422
7E	6E	3C	001D5	MOVZWL	CHANNEL, -(SP)	: 0435
67	01	FB	001D8	CALLS	#1, SYSSDASSGN	:
52	50	D0	001DB	MOVL	RO, STATUS_6	:
	08	AE	9F 001DE	PUSHAB	EFN	: 0436
00000000G	00	01	FB 001E1	CALLS	#1, LIB\$FREE EF	:
	0A	50	E9 001E8	BLBC	STATUS_7, 18\$	: 0437
	04	52	E8 001EB	BLBS	STATUS_6, 17\$	: 0438
	50	52	D0 001EE	MOVL	STATUS_6, RO	:
		04	001F1	RET		:
	50	53	D0 001F2	MOVL	FINAL_STATUS, RO	: 0444
		04	001F5	RET		: 0445

; Routine Size: 502 bytes, Routine Base: _LIB\$CODE + 0000

LIB\$SET_FILE_PRO LIB\$SET_FILE_PROT - Set file protection
V04-000 LIB\$SET_FILE_PROT - Set file protection

I 6
16-Sep-1984 02:26:57 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:34:28 [VMSLIB.SRC]LIBFILPRO.B32;1

Page 13
(4)

: 449 0446 1 END
: 450 0447 0 ELUDOM

! End of module LIB\$SET_FILE_PROT

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	502	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	89	0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS\$:LIBFILPRO/OBJ=OBJ\$:LIBFILPRO MSRC\$:LIBFILPRO/UPDATE=(ENH\$:LIBFILPRO)

: Size: 502 code + 0 data bytes
: Run Time: 00:13.9
: Elapsed Time: 00:16.3
: Lines/CPU Min: 1936
: Lexemes/CPU-Min: 24173
: Memory Used: 204 pages
: Compilation Complete

0436 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY